

REMARKS

The Office action has been carefully considered. The Office action rejected claims 1-35 and 40-64 under 35 U.S.C. § 112, first paragraph as failing to comply with the written description requirement. Further, the Office action rejected claims 36-39 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,833,840 to Lifshitz et al. ("Lifshitz"). Regarding the claim rejections, applicants respectfully disagree.

By present amendment, claims 1 and 40 have been amended. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on April 12, 2006. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a new approach to computer graphics that utilizes a new element object model and a vector graphics markup language for accessing element object models in a manner that allows program code developers to consistently interface with a scene graph data structure to produce

graphics. This new system and method includes a scene graph that may have objects that comprise an object model with associated application program interfaces (API). Having the API allows programmers to accomplish possibly complex composition effects within their applications in a straightforward manner, while leveraging the graphics processing unit in a manner that does not adversely impact normal application performance.

One aspect of the present invention is generally directed to an architecture, referred to as the media integration layer (MIL), that may include an immediate mode graphics application programming interface (API), a screen-partitioning data structure, a set of control level objects, and a markup language. In general, the architecture may allow program code, such as an application or operating system component, to communicate drawing instructions and other information (e.g., image bitmaps) to graphics components in order to render graphical output on the system display. As such, the present invention may provide a number of defined functions and methods, e.g., in the form of APIs to an object model, that enable programs to populate a scene graph with data structures, instruction lists (e.g., containing drawing primitives / commands), and other graphics-related data. When processed, the scene graph results in graphics being displayed on the screen.

Via the interfaces, the MIL may provide access to a data structure for storing visual information so that applications can take advantage of the graphics capabilities provided by the computer hardware. The interfaces may support an element object model and a vector graphics markup language for using that element object model in a manner that allows program code developers to

consistently interface with a scene graph data structure to produce graphics. The data structure may also be used for either directly rendering or for "compiling" the visual information so that it can be provided to a lower level graphics system for fast composition and animation.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

Rejections under §112

The Office action rejected claims 1-35 and 40-64 as being not supported by the written description. More specifically, the Office action contends that the recitation, in a markup language that is in a native format, is not contained in the specification. Applicants respectfully disagree and point out the specific teaching of this concept, and as discussed below, have changed the terminology to recite an "original" format.

In general, and as discussed above, the present invention is directed to a new approach to computer graphics that utilizes a new element object model and a vector graphics markup language (*i.e.*, a markup language in an original format) for accessing element object models in a manner that allows program code developers to consistently interface with a scene graph data structure to produce graphics. This new system and method includes a scene graph that may have objects that comprise an object model with associated application program interfaces (API). Having objects with APIs allows programmers to accomplish possibly complex composition effects within their applications in a straightforward

manner, while leveraging the graphics processing unit in a manner that does not adversely impact normal application performance.

More specifically, the vector graphics markup language (as well as other languages such as C#) may make originate calls to an API. Unlike systems of the past, any graphics language would necessarily need to be converted to another format in order to make a call to an API. Thus, the commands and calls within a conventional graphics language may be coded in an original (or “native”) format, but before being used within a graphics system of any kind, were required to be translated to another suitable format (e.g., pre-compiled) for calling an API. Therefore, the actual code that is used to call functions of an API does not retain its original format.

This concept of calling APIs without necessarily changing (e.g., pre-compiling) from an original format as generally described in the present application is found in the specification, starting on page 16. Example XAML code snippets in an original format that accomplishes the calls are shown throughout the specification, including on pages 24-28. For example, following the example code on page 24, the application states that “Using the Visual API, developers can instead draw directly into the Visual (that would otherwise be accessed via by the layout element).”

In this disclosure, it teaches that one aspect of the present invention is generally directed to allowing program code to communicate drawing instructions to graphics components in order to render graphical output on the system display. To this end, the present invention provides a number of defined functions and

methods, *e.g.*, in the form of application programming interfaces (APIs) to an object model, that enable programs to populate a scene graph with data structures, drawing primitives (commands), and other graphics-related data. When processed, the scene graph results in graphics being displayed on the screen.

The specification further teaches that FIG. 2 represents a general, layered architecture into which the present invention may be implemented. As represented in FIG. 2, program code (*e.g.*, an application program or operating system component or the like) may be developed to output graphics data in one or more various ways, including via imaging, via vector graphic elements, and/or via function / method calls placed directly to a visual application programming interface (API) layer. That is, the vector graphic elements and function/method calls are generated directly from an application and placed directly to an API. There is no previous translation step necessary, as the vector graphic elements and function/method calls are in their native and original format. Thus, despite the fact that the words “native” and/or “original” do not explicitly appear in the text of the specification, it is clear that the common usage of the terms native and original may be used to describe the nature of the code being described. Applicants submit that the phrase, “markup language in an original format,” finds support in the specification at least at the above-identified passages and that the rejections under §112, first paragraph, be withdrawn.

Rejections under §102

Turning to the §102 rejections of the claims, claim 36 generally recites in a computing environment, a computer system comprising a scene graph data structure containing data that can be rendered into integrated output that can be viewed, an object model including visual objects having an application program interface and other data that can be contained in the scene graph data structure, and a graphics interface operable to facilitate the scene graph data structure.

The Office action rejected claim 36 as being anticipated by Lifshitz. Specifically, the Office action contends that Lifshitz teaches a scene graph data structure containing data that can be rendered into integrated output that can be viewed. Fig. 3 of Lifshitz is referenced. Further, the Office action contends that Lifshitz teaches an object model including visual objects and other data that can be contained in the scene graph data structure. Again, Fig. 3 of Lifshitz is referenced. Further yet, the Office action contends that Lifshitz teaches a graphics interface operable to facilitate the scene graph data structure. Fig. 2 and modules 202, 204 and 200 of Lifshitz is referenced. Applicants point out that the Office action has not addressed a particular recitation of claim 36; specifically, the Office action does not address “a visual object having an application program interface” as clearly recited in claim 36. For this and other reasons, applicants respectfully disagree with the rejection.

Lifshitz is directed, generally, to a system for handling visual objects in the context of an MPEG-4 layering architecture. Specifically, the cited and applied section of Lifshitz details a conventional system that suffers from the very problem

that the present invention solves. The Office action even quoted the very section that teaches the problem, namely, that Lifshitz teaches “Modules 202 (an application) and 204 (network stacks) interact with the core module 200 through APIs.” Having such an application that uses an API for interaction with a graphics system causes the very problem that is overcome by having a visual object with an application programming interface, as recited in claim 36.

Thus, in perfect contrast with this statement cited from Lifshitz, claim 36 recites an object model including visual objects having an application program interface and other data that can be contained in the scene graph data structure. Lifshitz does not disclose utilizing any type of object having an application program interface within an object that is part of an object model. Note that the PROTO object of Lifshitz cannot be a visual object of the object model as specified in claim 36, as Lifshitz states that “once the PROTO object has been instantiated, it must be accessed from within the Renderer Plane.” Lifshitz column 7, lines 12 and 13 (emphasis added). The Office action left this recitation of claim 36 unaddressed, which it cannot do when making a §102 rejection. Applicants submit that claim 36 is allowable over the prior art of record for at least these reasons.

Applicants respectfully submit that dependent claims 37-39, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 36 and consequently includes the recitations of independent claim 36. As discussed above, Lifshitz fails to disclose the recitations of claim 36 and therefore these claims are also allowable over the prior art of record. In addition to

the recitations of claim 36 noted above, each of these dependent claims includes additional patentable elements.

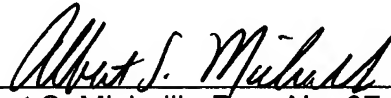
For at least these additional reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and timely allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-64 are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

CERTIFICATE OF MAILING

I hereby certify that this Amendment and RCE transmittal, along with and Credit Card Payment Form are being deposited with the United States Postal Service on the date shown below with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

Date: May 9, 2006



Albert S. Michalik

3471 third amendment